IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 11, 2019, accepted December 23, 2019, date of publication January 1, 2020, date of current version January 10, 2020.

Digital Object Identifier 10.1109/ACCESS.2019.2963487

Depth Map Estimation for Free-Viewpoint Television and Virtual Navigation

DAWID MIELOCH, OLGIERD STANKIEWICZ, (Member, IEEE), AND MAREK DOMAŃSKI, (Member, IEEE)
Chair of Multimedia Telecommunications and Microelectronics, Poznań University of Technology, 60-965 Poznań, Poland

Corresponding author: Dawid Mieloch (dawid.mieloch@put.poznan.pl)

This work was supported by the Ministry of Science and Higher Education of Republic of Poland.

ABSTRACT The paper presents a new method of depth estimation, dedicated for free-viewpoint television (FTV) and virtual navigation (VN). In this method, multiple arbitrarily positioned input views are simultaneously used to produce depth maps characterized by high inter-view and temporal consistencies. The estimation is performed for segments and their size is used to control the trade-off between the quality of depth maps and the processing time of depth estimation. Additionally, an original technique is proposed for the improvement of temporal consistency of depth maps. This technique uses the temporal prediction of depth, thus depth is estimated for P-type depth frames. For such depth frames, temporal consistency is high, whereas estimation complexity is relatively low. Similarly, as for video coding, I-type depth frames with no temporal depth prediction are used in order to achieve robustness. Moreover, we propose a novel parallelization technique that significantly reduces the estimation time. The method is implemented in C++ software that is provided together with this paper, so other researchers may use it as a new reference for their future works. In performed experiments, MPEG methodology was used whenever possible. The provided results demonstrate the advantages over the Depth Estimation Reference Software (DERS) developed by MPEG. The fidelity of a depth map, measured by the quality of synthesized views, is higher on average by 2.6 dB. This significant quality improvement is obtained despite a significant reduction of the estimation time, on average 4.5 times. The application of the proposed temporal consistency enhancement method increases this reduction to 29 times. Moreover, the proposed parallelization results in the reduction of the estimation time up to 130 times (using 6 threads). As there is no commonly accepted measure of the consistency of depth maps, the application of compression efficiency of depth is proposed as a measure of depth consistency.

INDEX TERMS Depth map estimation, free-viewpoint television, FTV, virtual navigation, multiview stereo, segmentation.

I. INTRODUCTION

In free-viewpoint television (FTV) and Virtual Navigation (VN) [29], [41], on which we focus in this paper, a user can arbitrarily change her/his viewpoint at any time and is not limited to watch views acquired by cameras located around a scene. Views presented to the user are synthesized, i.e., rendered using a compact representation of a 3D scene [38].

Nowadays, the most commonly used spatial representation of 3D scenes are depth maps [39], which are widely used not only in the context of free-viewpoint television and virtual navigation [1], [29], [40], but also in 3D scene modeling [36], and machine vision applications [37], [50]. In FTV and VN systems, the fidelity and quality of depth maps deeply influence the quality of the synthesized video, thus the quality of experience in the navigation through a 3D scene.

Real-time depth acquisition using depth cameras seems to be very attractive [4]. Nevertheless, the usage of depth cameras, or in general depth sensors, is hampered by their high cost, low resolution, limited measurement range, and interferences between cameras [5]. Moreover, depth sensors illuminate a scene by infrared light, which could be unacceptable in many applications. The abovementioned problems limit the possible applications of depth sensors in FTV and VN systems, although depth cameras and lidars have

The associate editor coordinating the review of this manuscript and approving it for publication was Li He.

recently undergone many improvements [58], [59]. Thus, the considerations of this paper are focused on depth estimation by multiview video analysis.

In FTV and VN, the estimation of depth maps is not the final goal, but it is rather an important step in the process of preparing the virtual views. Therefore, through this paper, the quality of the depth maps is represented by the quality of the virtual views synthesized using these depth maps. Such an approach is common in research on depth map estimation [61], [62], and was also proposed as a part of the 3D framework of the ISO/IEC MPEG group [60].

As it is discussed in Section II, although the methods described in the references provide satisfying quality for many applications, they are not well-matched to the needs of FTV and VN. In order to provide a very realistic viewing experience during virtual navigation, a new method of depth estimation has to meet a set of requirements that result from the characteristics of FTV and VN.

FTV is characterized by a high number of cameras used for multiview video acquisition. Moreover, the resolution of cameras used in multiview systems constantly increases, especially for new virtual reality systems [30], [57]. At the same time, depth estimation is already one of the most complex parts of multiview video processing in FTV/VN systems, therefore, achieving higher quality comes at the cost of a further increase of complexity.

The characteristics of depth estimation for FTV/VN require not only to reduce the high complexity of estimation but also to ensure inter-view and temporal consistencies of depth maps in order to avoid annoying artifacts in the synthesized video, such as, e.g., flickering and false reconstructions of objects. Virtual view synthesis uses depth maps and views from at least two nearest cameras [15], [38], [48]. The inter-view inconsistency of depth maps is related to independent depth estimation in neighboring views. Such independent estimation can cause inconsistency in the position of this object in a synthesized virtual view, which reduces both the objective and subjective quality of the synthesized view [3]. The temporal consistency of depth maps, on the other hand, means that the values of depth in consecutive frames of depth maps change in accordance with the movement of objects in a scene, and what follows, the color and position of objects in a virtual view also change in accordance with their movement.

The variety of hitherto presented FTV systems [42] makes it difficult to develop a versatile depth map estimation method that could be successfully utilized in all such systems. FTV/VN systems vary in the number and type of used cameras (from a few to hundreds), distances between them, and their positioning. Therefore, summarizing the requirements for FTV/VN systems, a new method for depth estimation should be characterized by the following features.

1) High quality of estimated depth maps, with particular emphasis on inter-view and temporal consistencies.
2) Versatility of estimation process, i.e. no assumptions about the number and positioning of cameras can be imposed, and moreover, the method can be used for different scenes without any modifications.
3) Processing time of estimation that is reduced in comparison to the state-of-the-art methods that meet the abovementioned requirements (e.g., for the new presented method variants of parallel implementations are studied).

The novelty of the proposed method consists in addressing the abovementioned characteristics by joint application of many ideas, e.g., the use of image segmentation, depth estimation performed simultaneously for all views, the cost function for improved inter-view consistency, the enhancement of temporal consistency, and also the utilization of parallelization. The details of the proposed method are presented in Section III. The novelty of this paper consists in:

- original segment-based depth estimation proposed by the authors – a less efficient version of the method was briefly described in our previous work [32],
- the novel temporal enhancement that significantly improves the temporal consistency of depth maps and decreases the processing time,
- the novel parallelization method for graph-based depth estimation methods,
- thoughtful experimental analysis and assessment of the proposed method.

II. STATE-OF-THE-ART- DEPTH ESTIMATION METHODS

The simultaneous fulfillment of requirements concerning the inter-view and temporal consistencies of depth maps, and at the same time, achieving a relatively short processing time of estimation, is difficult without compromising the quality of the estimated depth [18]. For example, independent estimation of depth maps for each camera can be faster than simultaneous estimation for all views [2], [26], however, the lowered number of views used during estimation causes the loss of inter-view consistency. Depth estimation can also be performed for input views with reduced resolution, nevertheless, the usage of low-resolution views decreases the accuracy of estimated depth maps and the resulting virtual view quality [33]. The loss of quality is especially visible near the edges or in highly textured regions [3], [19]. Even if additional depth refinement is applied in post-processing [34], [54], for depth maps estimated in low resolution, the quality is still lower than for the estimation for the original resolution, even if virtual view synthesis methods designed for low-resolution depth maps are used [35]. Method [70] consists in an iterative approach to deal with the low resolution of depth maps using depth refinement by the joint view synthesis and depth estimation.

Depth estimation based on stereoscopic correspondence is very time consuming, especially for global estimation methods that can provide depth maps of sufficient quality for view synthesis purposes (e.g., [7], [23], [28]). Nevertheless, such methods often require input views to be rectified or are designed for multi-view systems of different characteristics than FTV systems, e.g. for light-field systems

[21], [53], [55], or multi-camera arrays [49], which have much smaller distances between cameras. Inter-view and temporal consistencies are often also ensured, e.g. in [6] or [27], nevertheless, only for sequences acquired using a moving camera rig.

The use of depth estimation methods based on local estimation can ensure low complexity. Local estimation methods are very often suitable for real-time applications [18]. Although depth maps of relatively high quality [2], or even depth maps that are inter-view and temporally consistent [24] can be estimated using such methods, the majority of these methods formulate additional requirements about the number or positioning of cameras. For example, methods [24] and [26] can be used only for a stereo pair, while [2] and [4] are strictly adjusted to multiview systems designed by their authors, reducing the usefulness of these methods in versatile free-viewpoint television systems.

Depth maps can also be estimated using an epipolar plane image [8], [31]. These methods force depth to be consistent in all views and are characterized by lower complexity than global estimation methods, but can be used only for dense multi-view systems. More recently, a new interesting type of depth estimation methods was introduced, which uses convolutional neural networks to support the estimation process on the basis of a previously prepared database of depth maps. Such data-driven estimation, although it can represent the direction of future research in depth estimation, is still limited to specific applications (e.g., for soccer stadiums footage [22]), stereo pairs [16] or multi-view systems with a very narrow base [17], just like conventional methods presented earlier.

In order to shorten the processing time of estimation, depth optimization is often based on segments of input views, instead of on individual points, like in [63]. In this method, the inter-view correspondence is based on the matching of segments and the smoothness of estimated depth maps is proportional to the length of the boundary between neighboring segments. The use of image segmentation helps reduce the complexity of depth estimation and decreases the errors of estimation that are the result of poor representation of the edges of objects in point-level estimation. Nevertheless, the matching of segments in neighboring views is effective only when cameras are close to each other, because in sparse FTV systems the segmentation of the same object may be significantly different in neighboring views.

Other methods that employ image segmentation [23], [69] use the smoothing cost that is calculated between neighboring points of an image and the data cost calculated both for points and segments, and has been shown to achieve very good results in terms of the quality of depth maps. The method [65] utilizes graph-based depth estimation performed on the segments of input views, enhanced with the use of edge detection and plane matching. Nevertheless, the processing time for high-resolution stereo pair images for both methods is still calculated in a few minutes, moreover, inter-view consistency is not ensured.

In [26], image segmentation is used only in the correspondence search. The size of the matching window is large but limited by the boundaries of segments. It merges the advantages of large matching windows (limitation of the influence of noise) and small windows (possibility of correct depth estimation for small objects).

The use of segmentation in the depth map estimation process is widespread. What distinguishes the depth estimation method proposed by the authors, is that depth optimization in the proposed method is based only on segments of an image. In the presented state-of-the-art methods, optimization is also sometimes performed on segments, but at some step of estimation, point level optimization or further refinement is still required.

The temporal consistency of depth maps is often achieved through the use of additional refinement [62], [66]. Such refinement methods are usually based on the estimation and segmentation of the background of a scene. Unfortunately, the temporal consistency of objects in the foreground is not increased. The temporal consistency can also be increased with the denoising of input views used further in depth estimation [64]. The main advantage of such an approach is that denoising can be performed independently from depth estimation, therefore, it can be used with all depth estimation and refinement methods. On the other hand, an additional step of estimation increases the overall processing time.

Contrary to the abovementioned methods, the new temporal consistency enhancement of depth maps, presented by the authors in Section III-F, simultaneously decreases the complexity of the depth estimation process.

III. PROPOSED GRAPH-BASED MULTIVIEW DEPTH ESTIMATION METHOD

A. OVERVIEW OF THE PROPOSED METHOD

In the proposed approach, depth estimation is viewed as a recursive process, where frames from all real views are at the input. At a time instant, the output consists of depth maps for a number of views, i.e., using multiple input views, the number of depth maps are estimated for the consecutive time instants. The process of depth estimation is recursive in the sense that depth maps from previous time instants are used for the estimation of depth maps at the current time instant.

The novelty of the presented method of depth estimation, and its particular usefulness for free-viewpoint television and virtual navigation systems, is a result of the joint exploitation of the ideas mentioned below.

1) Depth is estimated for segments instead of individual pixels, and thus the size of segments can be used to control the trade-off between the quality of depth maps and the processing time of estimation. Larger segments can be used to attain fast depth estimation, or finer segments can be used to attain higher quality,
2) Object boundaries are collocated with segment borders, therefore segment-based depth estimation usually does not reduce depth map resolution.

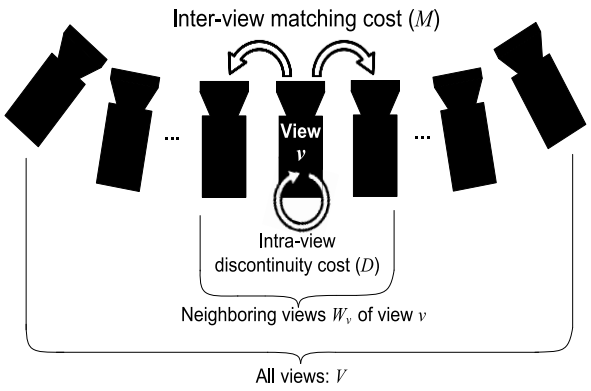

FIGURE 1. Views and cost function components used in the proposed depth estimation.

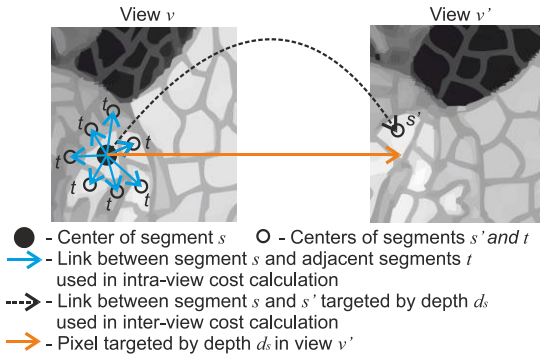

FIGURE 2. Visualization of intra-view discontinuity cost and inter-view matching cost for an exemplary segment *s* for depth estimation performed for 2 views.

3) Estimation is performed for all views simultaneously and produces depths that are inter-view consistent because of the utilization of the new formulation of the cost function, dedicated for segment-based estimation.
4) No assumptions about the positioning of views are stated: any number of arbitrarily positioned cameras can be used during the estimation.
5) Although segmentation is used, the estimated depth for each segment is calculated on a per-pixel basis, because the correspondence search is not limited to segment centers; the proposed method does not require the segmentation to be consistent in all views, therefore, it is performed independently in each view, reducing the overall complexity.
6) In the proposed temporal consistency enhancement method, depth maps estimated in previous frames are utilized in the estimation of depth for the current frame, increasing the consistency of depth maps and simultaneously decreasing the processing time of estimation.
7) The proposed depth estimation framework uses a novel parallelization method that significantly reduces the processing time of graph-based depth estimation.

B. COST FUNCTION FORMULATION

In the proposed method, depth estimation is based on cost function minimization. The proposed cost function is described over a set of views V (Fig. 1) for all of which depth maps are estimated.

There are two cost function components:

1) The intra-view discontinuity cost D, a smoothing, regularization term, defined inside each individual/particular view $v \in V$.
2) The inter-view matching cost M, responsible for the inter-view consistency of depth maps, defined between view v and each neighboring view $v' \in W_v$, where W_v is the neighborhood of the view v, e.g., the nearest left view and the nearest right view of the view v, whenever available.

In our approach, the cost function is defined with the use of segments, instead of individual pixels. For this, the segmentation is performed at the beginning, so that a set of segments S_v is attained independently for each view $v \in V$ (more details about the used segmentation technique can be found in Section IV-C). Therefore, the cost function components are defined as follows (Fig. 2):

1) The intra-view discontinuity cost, marked as $D_{s,t}$, penalizes depth discontinuities between two segments: $s \in S_v$ and segment t in neighborhood T_s of segment s in the same view v: $t \in T_s \subset S_v$.
2) The inter-view matching cost, marked as $M_{s,s'}$, penalizes dissimilarities between segments $s \in S_v$ and $s' \in S_{v'}$ that are matched by the currently considered depth map in views $v \in V$ and $v' \in W_v$ respectively.

Those two components, which are described in detail in Sections III-C and III-D, are used in the formulation of the overall cost function:

$$E(\underline{d}) = \sum_{v \in V} \sum_{s \in S_v} \{ \sum_{v' \in W_v} M_{s,s'}(d_s) + \sum_{t \in T_s} D_{s,t}(d_s, d_t) \}, \quad (1)$$

where $\underline{d}$ is a vector of depth values for all segments in all views, d_s is the depth of segment s (currently considered in vector $\underline{d}$), $v \in V$ are views for which depth is estimated, $v' \in W_v$ are views neighboring to view v, $s \in S_v$ are segments of view v, s' is a segment in view v' which corresponds to segment s in view v for depth d_s, $M_{s,s'}$ is the inter-view matching cost between segments s and s', $t \in T_s$ are segments neighboring to segment s, $D_{s,t}$ is the intra-view discontinuity cost between segments s and t, d_t is the currently considered depth of segment t.

It should be noted that the matching of segments between the views is done using depth d_s and can change during the estimation process (e.g., in consecutive iterations of the graph cut optimization algorithm). Therefore, for a given segment s, any depth value d_s can be selected, pointing at any pixel in view v' (presented as the orange arrow in Fig. 2), not necessarily a segment itself (or, e.g., its center, as presented in Fig. 2 with the dotted arrow).

C. INTRA-VIEW DISCONTINUITY COST

The intra-view discontinuity cost is calculated between all adjacent segments within a view (presented as the blue solid arrows in Fig. 2). The cost is calculated as follows:

$$D_{s,t}(d_s, d_t) = \beta \cdot |d_s - d_t|, \tag{2}$$

where β is a smoothing coefficient, d_s and d_t are the currently considered depths of adjacent segments s and t. The smoothing coefficient β is calculated adaptively using β_0, which is the initial smoothing coefficient provided by the user, and the similarity of segments s and t – the L1 distance (depicted as $\|\cdot\|_1$) between vectors $[\hat{Y}\hat{C}_b\hat{C}_r]_s$ and $[\hat{Y}\hat{C}_b\hat{C}_r]_t$ of average Y, C_b and C_r color components of the abovementioned segments:

$$\beta = \beta_0 / \left\| [\hat{Y}\hat{C}_b\hat{C}_r]_s - [\hat{Y}\hat{C}_b\hat{C}_r]_t \right\|_1, \tag{3}$$

therefore, when the similarity of adjacent segments is low, the smoothing coefficient also adaptively drops in value and thus the depths of these segments are not penalized for being discontinuous.

D. INTER-VIEW MATCHING COST

In order to achieve the inter-view consistency of estimated depth maps, the matching cost is not calculated independently for every single view. Instead, the conventional matching cost is replaced with the inter-view matching cost $M_{s,s'}(d_s)$, which is defined between a pair of segments s and s' that correspond to one another for the currently considered depth d_s (presented as the dotted arrow in Fig. 2).

The proper matching of whole segments from different views is a difficult operation. Moreover, for the presented method no assumptions about the positioning of views are made. Therefore, the segmentation of the same object in neighboring views may vary significantly, resulting in different shapes and sizes of the corresponding segments. These differences are especially big when the optical axes of cameras are not parallel, because corresponding parts of a scene can be visible from different angles in neighboring cameras. Inter-view consistent segmentation would require correct depth maps, obviously not available at the beginning of depth estimation.

In order to avoid the abovementioned difficulties, the inter-view matching cost is calculated in the pixel-domain in a small window A around the center of a segment and the corresponding point in a neighboring view. The core of the inter-view matching cost, denoted as $m_{s,s'}(d_s)$, is:

$$m_{s,s'}(d_s) = \frac{1}{size(A)} \sum_{a \in A} \left\| [YC_bC_r]_{\mu_s + a} - [YC_bC_r]_{T[\mu_s]+a} \right\|_1, \tag{4}$$

where A is a set of points in the window of the size specified by the user, a is a point in window A, $\|\cdot\|_1$ denotes L1 distance, μ_s is the center of segment s, $[YC_bC_r]_{\mu_s+a}$ is the vector of Y, C_b, C_r color components of the center μ_s of segment s, $T[\cdot]$ is a 3D transform obtained from intrinsic and extrinsic parameters of cameras, $[YC_bC_r]_{T[\mu_s]+a}$ is the vector of Y, C_b, C_r color components of the point in view v' corresponding to the center μ_s of segment s in view v.

In order to achieve inter-view consistent depth maps, the value of the inter-view matching cost $M_{p,p'}(d_p)$ is calculated as:

$$M_{s,s'}(d_s) = \begin{cases} min\{0, m_{s,s'}(d_s) - K\} & if\ d_s = d_{s'} \\ 0 & if\ d_s \neq d_{s'}, \end{cases} \tag{5}$$

where s is a segment in view v, d_s is the currently considered depth of segment s, s' is a segment in view v' which corresponds to segment $s$in view v for the currently considered depth d_s, $d_{s'}$ is the currently considered depth of segment s'. $M_{p,p'}(d_p)$ must decrease the value of the cost function when the compared segments have low inter-view matching cost, therefore, K must be a positive constant [44]. In the presented method, K presents a threshold for $m_{s,s'}(d_s)$, above which pair of segments s and s' is considered to be different objects and have assigned inter-view matching cost $M_{s,s'}(d_s) = 0$, therefore, the overall cost function $E(\underline{d})$ is not decreased. $m_{s,s'}(d_s)$ is an average difference between pixel values and, in idealistic case (without non-Lambertian reflections), s and s' should differ only by noise. Therefore, K can be assumed basing on noise existing in the images:

$$K \approx N_\sigma \cdot \sqrt{N_v} \cdot \sqrt{N_c} \cdot \sigma \tag{6}$$

In particular, we have decided to account for $N_\sigma = 5$ standard deviations of typical noise resulting from the aggregation of independent noise sources in the difference of two views ($N_v = 2$) and in the sum of three color components ($N_c = 3$). σ is a standard deviation of noise distribution existing in a single source. As it can be found in literature, for natural sequences σ can be up to 2.5 ([67], [68]). Basing on this, we obtained the value of $K = 30$ for the experiments.

The proposed inter-view matching cost makes the proposed method highly robust to the specular reflections on surfaces. Often, for sparse camera locations, like in FTV/VN systems, such specular reflection is visible in only one input view. For simplicity, assume that this reflection is visible in view v in segment r. This assumption simplifies notation but does not restrict the generality of the considerations. According to the abovementioned assumption, specular reflection highly increases the value of $m_{r,r'}(d_r)$ (4), where r' is a segment in view v' which corresponds to segment $r$in view v for depth d_r, $v' \in W_v$ are views neighboring to view v. In this case, $m_{r,r'}(d_r) > K$, therefore, the value of $M_{r,r'}(d_r) = 0$. The value of the cost $E(\underline{d})$ (1) for segment r becomes dependent only on the intra-view discontinuity cost $D_{r,q}$:

$$E(d_r) = \sum_{q \in T_r} D_{r,q}(d_r, d_q), \tag{7}$$

where d_r is the depth of segment r, T_r is the set of segments that neighbor to segment r, $D_{r,q}$ is the intra-view discontinuity cost between segments r and q, d_r is the currently considered depth of segment r and d_q is the currently considered depth of segment q.

$D_{r,q}$ is calculated using the similarity of adjacent segments r and q (2), therefore, the value of depth estimated for the described segment r is implied by the depth values estimated for similar adjacent segments in the same view. The proposed cost function decreases the influence of specular reflections on the final depth map quality. Also, the influence of other non-Lambertian reflections (i.e., direction-dependent) on the final quality of depth maps is limited, as the inter-view matching cost is defined only between the currently processed view and its closest left view, and the closest right view.

As a result of using the proposed cost function, depth is estimated also for disoccluded areas. Segments that represent parts of background objects that are visible only in one view (i.e., are occluded in other views), will also likely have a high value of (4), as they do not have the corresponding segment in another view. Again, because of the proposed formulation of the inter-view matching cost, the value of the overall cost (1) for disoccluded segments becomes dependent only on the intra-view discontinuity cost. Therefore, the value of depth estimated for the disoccluded segment is implied by the depth values estimated for similar adjacent segments, i.e., the depth values of background objects in the neighborhood of the disoccluded segment.

The presented definition of the inter-view matching cost does not require segmentation that is inter-view consistent in neighboring views. The center of a segment can correspond to any point in the neighboring view, not necessarily the center of a segment. Therefore, the presented pixel-domain matching lets us estimate the depth with high precision, simultaneously reducing the processing time of estimation, as the matching is not performed for all points, but only for centers of segments.

E. COST FUNCTION APPLICATION DETAILS

In the considered scenario, the optical axes of cameras do not have to be parallel. Therefore, in order to achieve inter-view consistency, the depth of a point has to be defined not as the distance from the plane of the camera that acquired this point, but as the distance from the plane of the center camera of the system [43] (for the sake of comprehension: the plane of a camera is a plane that contains the sensor of the camera).

A local minimum of the cost function $E\left(\underline{d}\right)$ (1) is estimated using the graph cut method [9] and the α-expansion method of minimization for multi-label problems, described in detail in [10]. At the beginning of the cost function $E\left(\underline{d}\right)$ minimization, we initially assume that all segments have the furthest possible value of depth for an actually processed scene, i.e., an approximate depth of the farthest object in the scene. In order to calculate the required depth of the furthest object, its approximate depth in real-world units (e.g. in meters) has to be converted back to the unit used in camera parameters. Such conversion can be easily calculated from camera parameters on the basis of the rough approximation of a distance between cameras of multi-view system. Nevertheless, such value is usually provided with multiview test sequences as z_{far} parameter (what was a case for all test sequences used in performed tests).

Unlike in [9], where each node in the constructed graph represents one point of an input view, in our method, a node corresponds to one segment. Nodes are connected by two types of links which correspond to abovementioned intra-view discontinuity cost and the inter-view matching cost (Fig. 2).

The proposed segment-based estimation reduces the number of nodes in a graph in comparison with point-level estimation, making the process significantly faster. Simultaneously, depth maps in the presented method are still estimated in the same resolution as the nominal resolution of the input views, and because of the use of segments, the edges of objects in depth maps correspond to the edges of objects in input views.

The number of segments, and therefore their size, is one of the estimation parameters and can be adjusted. The use of very small segments (i.e., of the size of 20 samples or less) allows us to estimate high-quality depth significantly faster than in pixel-based estimation. On the other hand, the use of larger segments ensures an additional reduction of the processing time, at the expense of a minor loss of quality (as proved by the tests of the influence of the number of segments on the virtual view quality – Section VI-A3).

F. TEMPORAL CONSISTENCY ENHANCEMENT

In natural video sequences, only a small part of an acquired scene considerably changes in consecutive frames, especially when cameras are not moving during the acquisition of video. The idea of the proposed temporal consistency enhancement of depth estimation is to calculate a new value of depth only for the segments that changed (in terms of their color) in comparison with the previous frame.

The proposed temporal consistency enhancement method allows us to automatically mark segments as unchanged in consecutive frames. These segments are used in the calculation of the intra-view discontinuity and the inter-view matching cost for other segments but are not represented by any node in the structure of the optimized graph. It reduces the number of nodes in the graph, making the optimization process significantly faster, and on the other hand, increases the temporal consistency of estimated depth maps.

In the first frame of a depth map, denoted as an ''I-type'' depth frame (by analogy to video compression terminology), the estimation is performed for all segments, as described in the previous sections. The following frames (''P-type'' depth frames) can utilize depth information from the preceding P-type depth frame and the I-type depth frame.

Segment s is marked by the algorithm as unchanged in two cases: if all components of the vector $[\hat{Y}\hat{C}_b\hat{C}_r]_s$ of average Y, Cb and Cr color components changed less than the set threshold T_b in comparison with the segment s_B, which is a collocated segment in the previous P-type frame, or, if all components of the abovementioned vector changed less than the threshold T_I in comparison with the segment s_I (a collocated segment in the I-type frame). If any of these

two conditions are met, then segment s adopts the depth from the segment s_B or s_I (depending on which condition was fulfilled).

A collocated segment in the previous or the first frame is simply the segment that contains the central point of the segment s. Therefore, even if the segmentation in compared frames is not the same, the algorithm can easily find the corresponding segment in these frames.

The introduction of two reference depth frames has a beneficial impact on the visual quality of virtual navigation in free-viewpoint television. First, the adoption of depth from the previous P-type depth frame allows us to use the depth of objects that changed their position over time. On the other hand, the adoption of depth from the I-type depth frame minimizes the flickering of depth in the background.

In the presented temporal enhancement, the average colors of whole segments are compared, therefore the influence of noise is much lower than in the inter-view matching cost (3), where points from input views are used. Therefore, the threshold was set to $T_P = 3$. In order to take into account the possible change in the scene illumination that could occur from the previous I-type depth frame, T_I should be lower than T_P. In all our tests we use $T_I = 1$.

IV. DEPTH ESTIMATION FRAMEWORK IMPLEMENTATION DETAILS

In this section, we present the methods and solutions used in our implementation of the proposed depth estimation.

A. PARALLELIZATION METHOD

In order to decrease the overall processing time of depth estimation in the presented method, the estimation is performed in parallel. In our proposal, each of n threads estimates a depth map with an n-times lower number of depth levels (depth levels are planes that are parallel to the plane of the camera). In the presented method, depth levels can be distributed onto threads in two ways: depth levels can be interleaved or divided into blocks (Fig. 3).

The distribution of depth levels has an influence on the processing time and quality of the estimated depth maps. If objects of an acquired scene are placed more densely in some ranges of depths, the estimation for corresponding depth levels is longer. Therefore, if the depth levels are divided into blocks, the estimation for some threads can be longer, increasing the overall processing time of depth estimation. On the other hand, when depth levels are interleaved, the processing time of estimation for all threads is nearly equal, but the estimated depths tend to be less smooth. The dependency between the type of parallelization and the performance of the depth estimation method was tested in one of the performed experiments presented in Section VI.

Depth maps with a reduced number of depth levels that were calculated by different threads have to be merged into one depth map. The merging process is performed in a similar

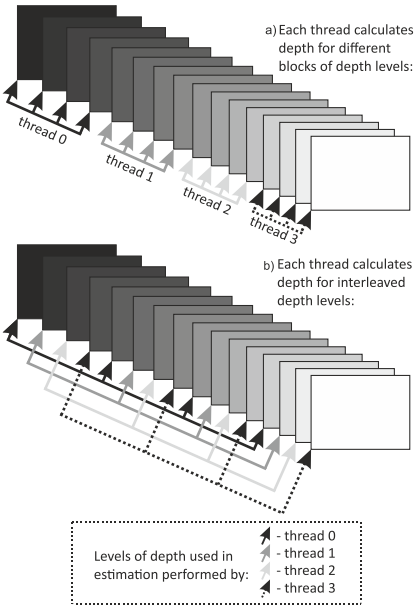

FIGURE 3. Two examples of different depth level distributions over threads in the proposed method: a) depth levels are divided into blocks, b) depth levels are interleaved. Each rectangle represents a different level of the depth of a scene.

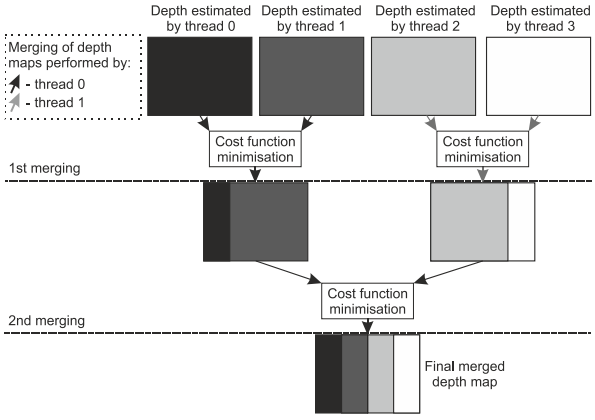

FIGURE 4. Depth map merging process for the 4-thread parallelization.

way as depth estimation [using the cost function (1)], but only two levels of depth are considered for each segment – i.e., the depth of a segment from thread t or the depth from thread $t+1$ (Fig. 4). Only two depth maps can be merged into one by one thread during the merging cycle. Therefore, for n threads, $\lceil log2(n) \rceil$ of additional cycles are needed to estimate the final depth map with all depth levels.

Of course, even without the use of parallelization, all cores of the CPU can also be used for depth estimation, e.g., each core can perform the estimation of depth for different sets of input views (e.g., for each 5 cameras of the system), or for different frames of the sequence. Unfortunately, when many standalone depth estimation processes are performed, it results in the loss of inter-view consistency or temporal consistency of estimated depth maps. When the proposed parallelization is used, both inter-view

and temporal consistency of depth maps, which are fundamental for the quality of virtual view synthesis, are preserved.

B. OPTIMIZATION METHOD

The proposed method utilizes the graph cuts method to estimate depth maps [9], [10]. As was proven in [25], the improvement of problem formulation has a significantly larger influence on depth estimation performance than the selection of the optimization method. Additionally, the graph cuts method, in comparison with belief propagation, the competitive method of global optimization, handles the penalties between nodes of the graph in a better way [25]. Therefore, in the proposed method of depth estimation, where graph construction is strictly based on dependencies between segments, the use of the graph cuts method is advisable and favorable.

C. SEGMENTATION

The proposed method of depth estimation can be used with any superpixel segmentation method. The authors decided to use the SNIC method (Simple Non-Iterative Clustering [20]) because the properties of SNIC meet the characteristics of the proposed depth estimation method: segments represent small regions, not whole objects, and the number of segments can be freely changed. The SNIC method has also been shown to have low complexity (which reduces the overall processing time of depth estimation) and achieve one of the lowest segmentation errors when compared to state-of-the-art methods, which positively influences the representation of edges of objects in depth maps.

In the presented framework of depth estimation, instead of the CIELAB space, in order to avoid the recalculation of color space, the segments are calculated using the YC_bC_r color space. The parameters of the segmentation used are the compactness factor $m = 5$, and 8-connected segments.

V. SOFTWARE IMPLEMENTATION OF THE METHOD

The above-described method is implemented as C++ software provided for use in further research. The software can be downloaded together with a manual, configuration examples, and license details from the following repository: https://gitlab.com/dmieloch/depth-map-estimation-for-ftv.

Currently, DERS is available for comparison, but the software for newer methods remains unavailable for a broader research community. Here, complementary software is provided for the convenience of the research community. The authors believe that the availability of this new software will be useful as an additional reference for future developments in depth estimation.

VI. EXPERIMENTAL RESULTS

A. ASSESSMENT OF THE QUALITY OF DEPTH MAPS

1) DESIGN OF EXPERIMENTS

In the experiments presented in whole Section VI-A, the quality of depth maps is measured indirectly, through virtual view synthesis. For an end user, the quality of virtual views expresses the overall quality of a free-viewpoint television system. Therefore, virtual views are a good determinant of the performance of a depth estimation method.

TABLE 1. Test sequences used in experiments.

Test sequence	Resolution	Used views	Sequence source
Ballet Breakdancers	1024×768	0 to 7	Microsoft Research [11]
BBB Butterfly BBB Rabbit	1280×768	6, 12, 19, 26, 32, 38, 45, 52	Holografika [12]
Poznań Blocks Poznań Blocks2 Poznań Fencing2 Poznań Service2	1920×1080	0 to 7	Poznań University of Technology [13][14]

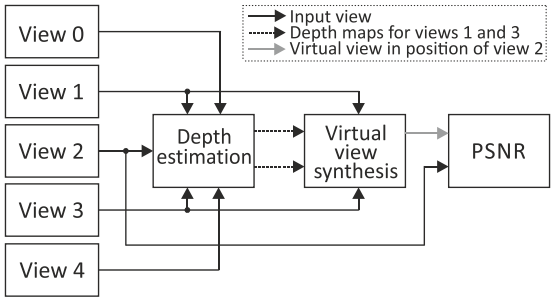

FIGURE 5. The scheme of PSNR calculations for the virtual view synthesized using depth maps estimated in the experiment.

In the experiments, a set of 8 multiview test sequences of varied character and arrangement of cameras are used. Sequences, their resolutions, views used in experiments and their sources are presented in Table 1.

In the conducted experiments not only do we compare our method with the state-of-the-art graph-based depth estimation method DERS [7] (Section VI-A2), but we also determine the performance of the presented method for different numbers of segments (Section VI-A3), and for different numbers of views used in the estimation (Section VI-A4). The performance of the presented parallelization methods and temporal consistency enhancement is also tested (Sections VI-A5 and VI-A6 respectively).

The scheme of measuring depth map quality is presented in Fig. 5. The synthesis of a virtual view placed in the position of the acquired view 2 is performed using neighboring views 1 and 3 and corresponding estimated depth maps. The synthesized virtual view is compared with the acquired view 2 and PSNR of luminance is calculated and averaged for 50 frames for each test sequence. In the experiments, besides the quality of estimated depth maps, we also measure the processing time of estimating depth per one frame and view of a sequence. There are 5 views used during estimation, except for the analysis of the influence of the number of views on the quality of virtual views (Section VI-A4). In order to decrease the overall processing time of the estimation, temporal consistency enhancement is turned on in all experiments (the number of P-type depth frames between I-type depth frames is equal to 9).

It is worth noting that in the case of free navigation, the virtual views are estimated from two or more nearest views, e.g., the virtual view between acquired views 1 and 2 is usually synthesized using exactly these two views. The nearest acquired view is, in the worst possible case, distant from the virtual view by half of the distance between cameras. The distance between the position of the virtual view and the acquired views has a significant impact on the quality of virtual views [3]. Here, the distance between views used for view synthesis is larger, therefore the overall quality of virtual navigation obtained from estimated depth maps would be noticeably higher than presented in the experiments.

All experiments were performed on one thread of Intel Core i7-5820K CPU (3.3 GHz clock) machines equipped with 64 GB of operational memory (except for the test of the parallelization method, where the number of used cores varied from 1 to 6). The size of a block in the inter-view matching cost is 3×3, the estimation is performed for 250 levels of depth and the initial smoothing coefficient is the same for both methods ($\beta_0 = 1$). The synthesis of virtual views is performed using the View Synthesis Reference Software developed by the MPEG community [15].

2) COMPARISON WITH DERS

The presented method is compared with the state-of-the-art Depth Estimation Reference Software developed by the MPEG community [7]. DERS is a graph-based method available for researchers in its entirety and it states no assumptions about the positioning of cameras. Therefore, DERS is a reasonable reference depth estimation method for the presented framework.

For HD test sequences (listed in Table 1) the number of segments used in the proposed method is 100,000, while for sequences with the lower resolution, in order to ensure a similar size of segments for all sequences, the number of segments is 50,000. Other parameters of estimation are the same for both methods.

Table 2 presents the results of the experiment. For all tested sequences the quality of virtual views synthesized using depth maps estimated with the proposed method is higher than for depth maps from DERS, with the maximum gain in quality equal to more than 5 dB. The average gain for all sequences is 2.63 dB. The lowest PSNR of a virtual view for DERS is below 22 dB, while for the proposed method the lowest PSNR is 25.5 dB. For the proposed method, only for one sequence the PSNR is below 27 dB. For DERS there are five such sequences. The visual comparison of depth maps for the proposed method and DERS, together with synthesized virtual views, is shown in Fig. 6 and is available in the video attached to this paper as supplementary material.

As Table 3 shows, the estimation process is, on average, more than 4 times faster for the presented method, even when the temporal enhancement and parallelization are not used. What is important, the reduction of the processing time of estimation is highest for HD sequences, therefore, the proposed method can be effectively used with high-resolution cameras. It is the effect of the use of segmentation in depth estimation – the complexity of estimation in the proposed method is dependent on the number of segments, not on the resolution of input views.

TABLE 2. Comparison of quality of virtual views synthesized using depth maps estimated using the proposed method and the reference method DERS [7].

Test sequence	PSNR of virtual view [dB]		
	DERS	Proposal	Gain
Ballet	27.93	**28.69**	0.76
Breakdancers	31.13	**32.19**	1.06
BBB Butterfly	29.97	**33.20**	3.23
BBB Rabbit	22.59	**27.21**	4.62
Poznań Blocks	21.97	**27.20**	5.23
Poznań Blocks2	25.67	**28.12**	2.45
Poznań Fencing2	26.74	**28.60**	1.86
Poznań Service2	23.69	**25.51**	1.82
		Average:	2.63

TABLE 3. Comparison of the processing time of the estimation of depth maps for the reference method DERS [7] and the proposed method with proposed enhancements.

Test sequence	Processing time of depth estimation per one view [s]			
	DERS (reference)	Proposal w/o enhancements	Proposal w/ temporal enhancement	Proposal w/ temporal enhancement and parallelization
Ballet	882.3	499.2	62.9	**11.9**
Breakdancers	949.1	254.8	41.1	**9.1**
BBB Butterfly	593.1	278.9	41.1	**7.0**
BBB Rabbit	744.9	91.8	16.3	**4.5**
Poznań Blocks	1445.8	313.1	50.4	**14.4**
Poznań Blocks2	1060.4	209.8	40.8	**10.2**
Poznań Fencing2	2254.2	391.2	60.9	**13.5**
Poznań Service2	2780.3	305.5	53.9	**12.7**
Average:	1338.8	293.1	45.9	**10.4**

3) RESULTS FOR DIFFERENT NUMBERS OF SEGMENTS

In the next experiment, the influence of the number of segments used in depth estimation on the quality of a virtual view is tested. The number of segments varied from 1,000 to 150,000.

The results of the experiment, averaged for all sequences, are presented in Fig. 7. As it can clearly be observed, the more segments are used in the estimation, the higher quality of depth maps can be achieved. However, the use of more than 100,000 segments insignificantly increases the quality of depth maps, at the cost of a considerable increase of estimation time.

When only 1,000 segments are used, the quality of depth maps is equal to the average quality of depth maps estimated using the DERS method, but the time needed for the estimation process is significantly shorter and equal to only two seconds.

The highest increase in the quality of depth maps can be seen between estimations performed for 1,000 and 25,000

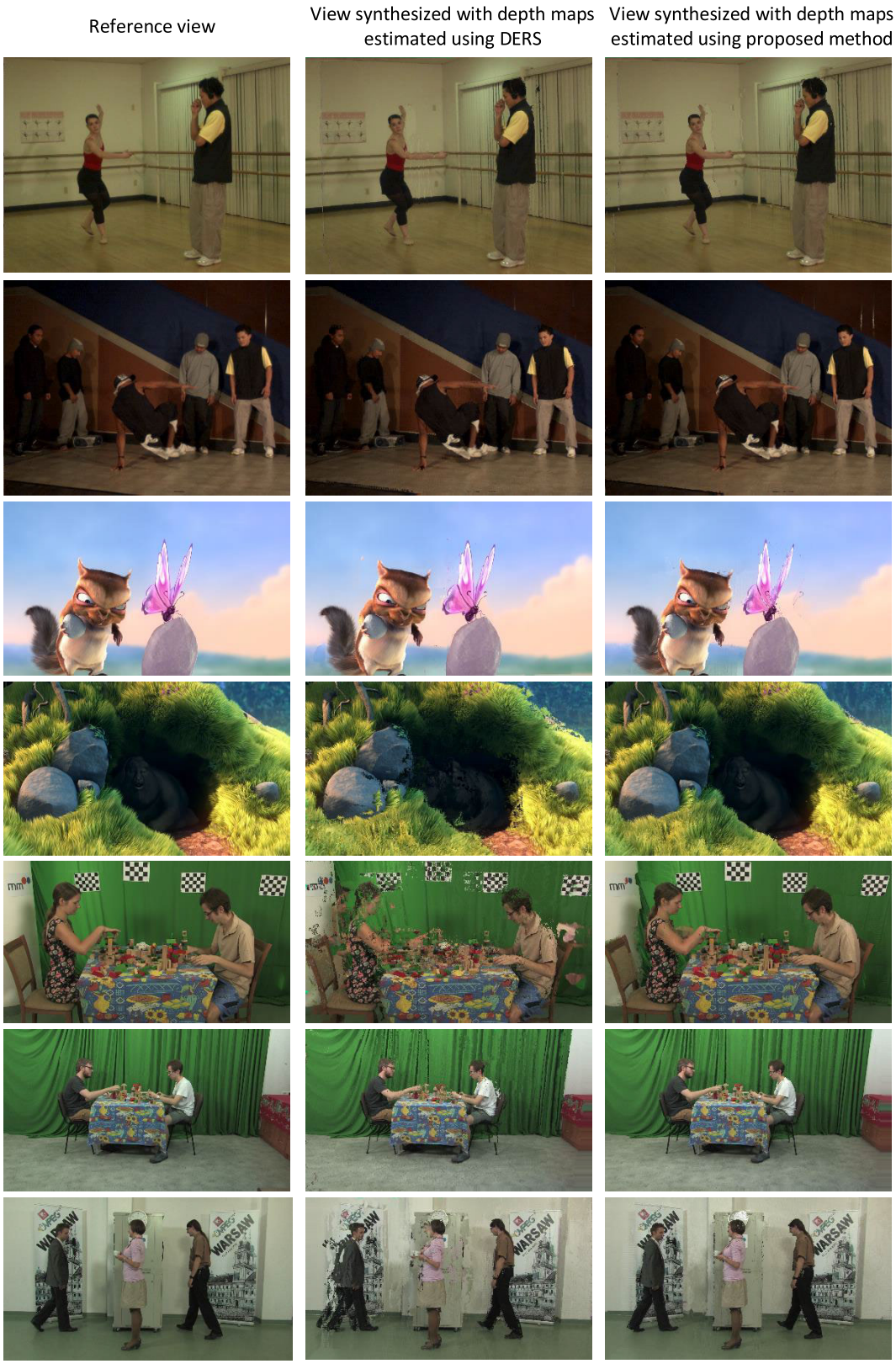

FIGURE 6. Comparison of virtual views synthesized using depth maps estimated using DERS and the proposed method.

segments per view. Despite the number of segments increasing 25-fold, the average processing time of estimation increases only six-fold. On the other hand, increasing the number of segments above 100,000 does not change the quality of depth maps significantly (only by 0.1 dB), but the mean processing time of estimation is noticeably longer.

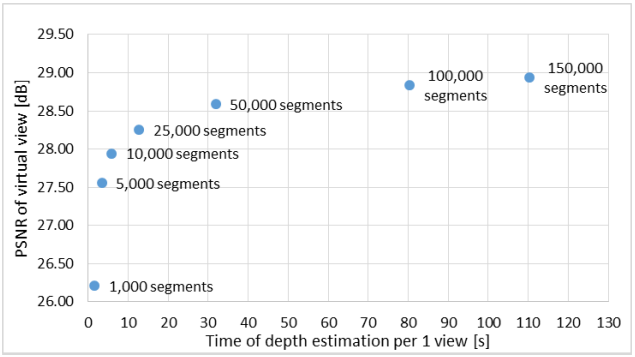

FIGURE 7. The average quality of a virtual view synthesized using depth maps estimated for different numbers of segments per one view and processing times of depth estimation.

The visual comparison for the Poznań Fencing2 sequence of depth maps estimated for different numbers of segments is presented in Fig. 8, while the virtual views synthesized using these depth maps are presented in Fig. 9. The comparison clearly shows a much better representation of edges of objects resulting from using segments instead of point-based estimation. The reduction of the number of superpixels, at the expense of a very minor loss of quality, gives a significant reduction of the required processing time.

The results for individual sequences are presented in Table 7 in the Appendix. The visual comparison of depth maps estimated for different numbers of segments, together with synthesized virtual views, is also available in the video attached to this paper as supplementary material.

4) RESULTS FOR DIFFERENT NUMBERS OF VIEWS

The influence of the number of views used in the estimation process on the quality of the estimated depth maps is also tested. The number of views varies from 3 to 8 and is limited by the number of views available in test sequences.

The results presented in Fig. 10 show that the use of more than 5 views changes the measured quality of virtual views and the processing time of estimation only to a small extent. However, the use of all available views increases the inter-view consistency of estimated depth maps, therefore, we recommend performing the estimation for all views simultaneously to ensure the high quality of free navigation. The results for individual sequences are presented in Table 8 in the Appendix.

5) RESULTS FOR DIFFERENT TYPES OF PARALLELIZATION

The presented parallelization method is tested in two variants: blocks of depth levels (Fig. 3a) and interleaved levels of depth (Fig. 3b). The number of used threads varies from 1 to 6 and is limited by the number of standalone cores in the used CPU. The results of the experiment (Fig. 11) confirm that if the levels of depth are distributed onto threads as blocks of depth levels, the processing time of the estimation is slightly longer than for interleaved levels of depth, but the difference in quality increases with the number of threads used.

Even when 6 threads are used, the quality decrease in comparison with the estimation without parallelization is insignificant (around 0.1 dB) but the processing time of the estimation decreases 4.5-fold. The results for individual sequences are presented in Table 9 in the Appendix. The visual comparison of depth maps estimated using the proposed parallelization method, together with synthesized virtual views, is also available in the video attached to this paper as supplementary material.

Moreover, both the inter-view and temporal consistency of depth maps, which are fundamental for the quality of virtual view synthesis, are preserved when the proposed parallelization is used. The method is fully scalable, so the constantly increasing number of cores in modern CPUs can be fully utilized for further reduction of the processing time of depth estimation.

6) RESULTS FOR DIFFERENT NUMBERS OF P-TYPE DEPTH FRAMES

Here, we present the performance of the proposed temporal consistency enhancement of the proposed depth estimation method. The number of frames is 50, as in all conducted experiments, and the number of used P-type depth frames between I-type frames varies from 0 to 49.

Fig. 12 shows the results of the performed experiment. The temporal consistency enhancement significantly reduces the processing time of estimation (10 times when only one I frame is used) with a negligible decrease of the objective quality (less than 0.3 dB). The results for individual sequences are presented in Table 10 in the Appendix. The visual comparison of depth maps estimated using the proposed temporal consistency enhancement method, together with synthesized virtual views, is also available in the video attached to this paper as supplementary material.

The results presented above only refer to the quality of virtual views and do not reflect the improvement of temporal consistency of depth maps. As it was presented earlier [45], the size of depth maps after encoding is one of the objective measures of their temporal consistency. In this article, we focus on the quality of free navigation for a user of the FTV system, therefore, in order to measure the increase of the temporal consistency of depth maps, synthesized virtual views are compressed with the HEVC encoder. The lack of temporal consistency of depth maps results in the visible flickering of a virtual view. Therefore, the lower the temporal consistency of depth maps, the lower the efficiency of the encoding of virtual views.

The encoder is set in the low-delay mode, so only the first frame of virtual views is encoded as an intra frame. Such settings of the encoder increase the influence of temporal consistency of the encoded sequence on the final bitrate. In the experiments, we use the HM 16.15 framework [46] using MPEG common test conditions (with the exception of used test sequences) and software reference configurations.

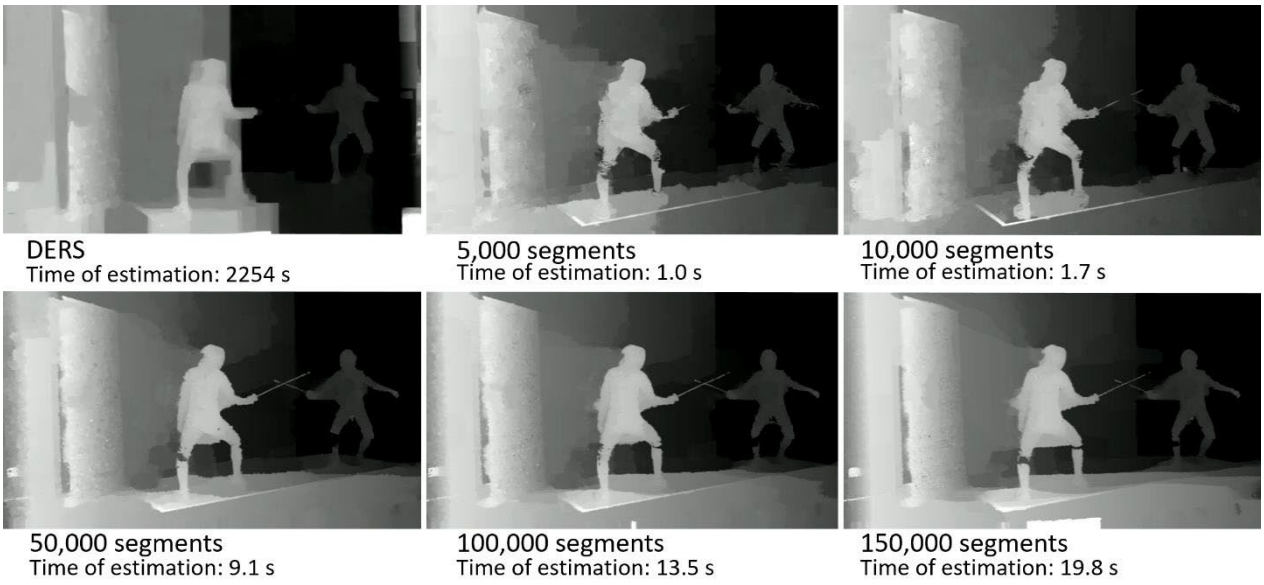

FIGURE 8. Comparison of depth maps (view #1 of Poznań Fencing2) estimated using the reference method DERS and the proposed method for different numbers of segments. The estimation times are given for one view of the sequence, averaged over 50 frames. For the proposed method the temporal enhancement is turned on and the estimation is calculated using 4 threads of CPU.

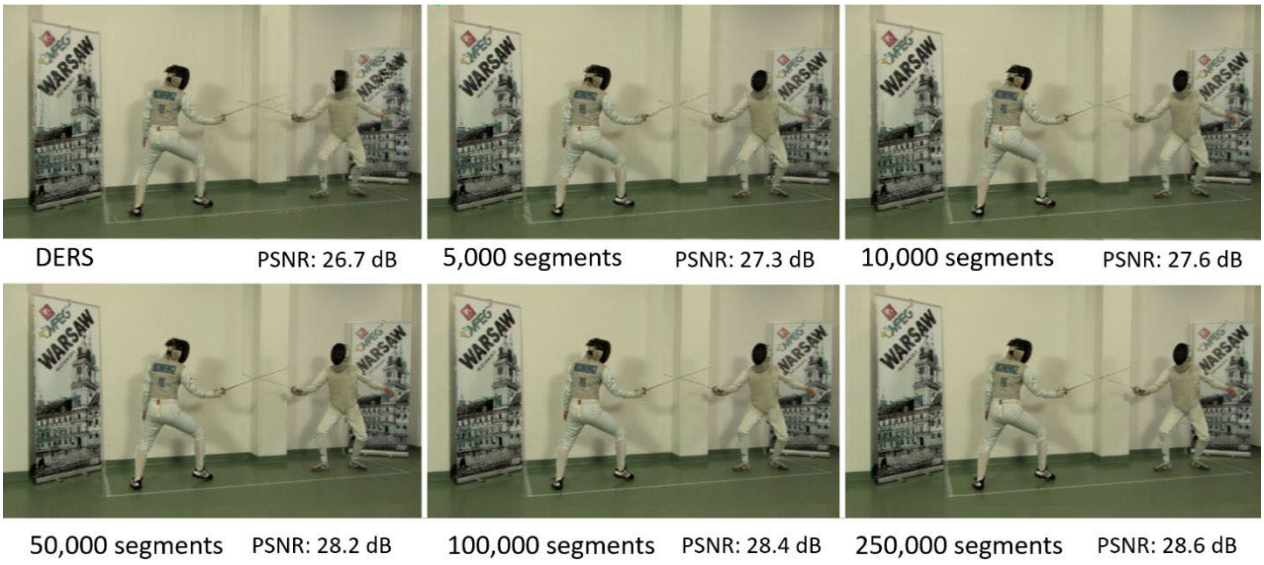

FIGURE 9. Comparison of virtual views (view #2 of Poznań Fencing2) synthesized for depth maps estimated using the reference method DERS and the proposed method for different numbers of segments. The PSNR values are derived with respect to the collocated input view.

Table 4 presents the results of encoding virtual views synthesized using depth maps with different numbers of P-type depth frames. The results are expressed as average luma bitrate reductions calculated using the Bjøntegaard [47] metric in comparison to a virtual view synthesized with depth that was not temporally enhanced. The detailed results for all QPs that include a bitrate and PSNR after encoding are presented in Table 11 in the Appendix.

For all tested sequences, the use of the proposed temporal consistency enhancement of depth maps results in bitrate reduction for all encoded virtual views. The average reduction is even higher than 30% when the number of P-type depth frames is equal to 49 (therefore only one I-type depth frame is used in the whole sequence). It indicates that the proposed technique of temporal consistency enhancement significantly increases the temporal consistency of depth maps, because the performance of the encoder in low-delay mode is vastly dependent on temporal prediction. The results also show another advantage of temporal consistency of depth maps in the FTV system – the reduction of the bitrate required to send a virtual viewpoint to an end user.

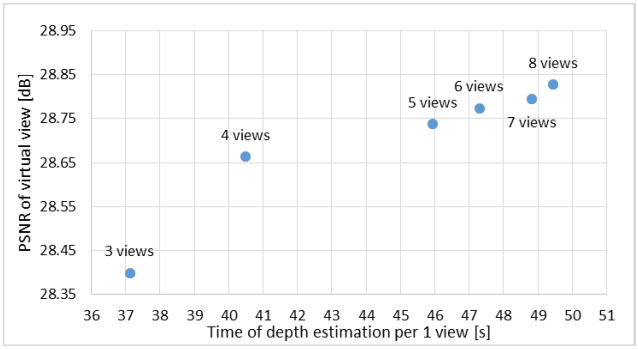

FIGURE 10. The average quality of a virtual view synthesized using depth maps estimated for different numbers of views used in the estimation process and processing times of depth estimation.

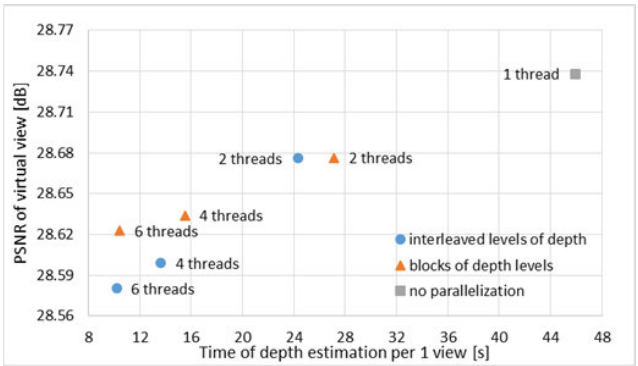

FIGURE 11. The average quality of a virtual view synthesized using depth maps estimated for different parallelization cases and processing times of depth estimation.

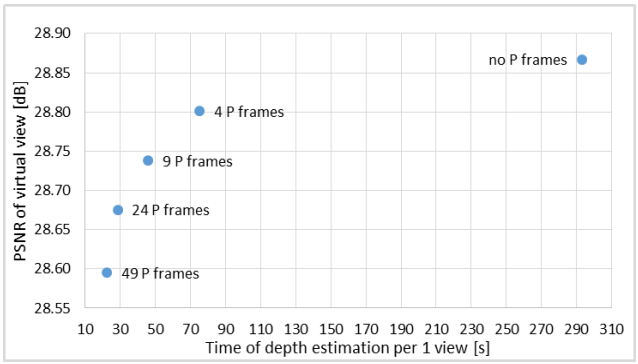

FIGURE 12. The average quality of a virtual view synthesized using depth maps estimated for different numbers of P-type depth frames between I-type depth frames and processing times of depth estimation.

B. ASSESSMENT OF THE ACCURACY OF DEPTH MAPS

The available databases with ground truth depth maps do not correspond to the characteristics of free-viewpoint television. The newest Middlebury database [51] is widely used by the research community and allows us to easily evaluate the performance of a depth estimation method and compare it with other methods. Unfortunately, the comparison of depth estimation methods in this database is performed for a set of rectified stereo-pair images acquired using two cameras with parallel optical axes, while in free-viewpoint television systems any number of arbitrarily positioned cameras can be used. Moreover, the dataset includes only one frame for each scene, therefore, the temporal consistency of depth maps, which is a significant part of the research presented in this paper, cannot be measured using this database.

Other databases of ground truth depth maps (e.g., one of the newest databases – the ETH3D Benchmark [52]) also focus on the use of multi-camera systems of different properties than FTV, e.g., on moving camera rigs, or on the 3D reconstruction of static scenes.

In order to provide direct evaluation of the accuracy of depth maps, we use the older Middlebury database [71], in which more views are available for some of the multiview images. In particular, we use two high-resolution (1800×1500) multiview images: Cones and Teddy, for which 9 views are available. Such a scenario to some degree meets the characteristics of the FTV and VN systems, therefore, it can provide fair quantitative results for the presented multiview depth estimation method.

TABLE 4. Average luma bitrate reductions of encoded virtual views synthesized using depth maps estimated for different numbers of P-Type depth frames between I-Type depth frames.

Test sequence	Number of P type depth frames between I-type depth frames			
	4	9	24	49
	Encoded virtual views bitrate reduction compared to virtual views synthesized using depth maps with no P-type depth frames			
Ballet	-14.4%	-19.0%	-20.5%	-20.8%
Breakdancers	-24.9%	-33.1%	-34.4%	-34.1%
BBB Butterfly	-18.8%	-29.2%	-34.6%	-38.3%
BBB Rabbit	-7.9%	-8.0%	-11.3%	-19.7%
Poznań Blocks	-4.5%	-5.7%	-7.3%	-6.1%
Poznań Blocks2	-30.7%	-35.0%	-36.9%	-39.9%
Poznań Fencing2	-30.7%	-53.5%	-68.8%	-75.8%
Poznań Service2	-21.6%	-30.0%	-23.5%	-21.7%
Average:	-19.2%	-26.7%	-29.7%	-32.0%

TABLE 5. The results of the assessment of the accuracy of depth maps estimated using the proposed method on the available 9 views high-resolution Middlebury dataset images [73].

Test images	Percentage of bad pixels		Average error	Average relative error	RMSE
	$E_T = 2.0$	$E_T = 4.0$			
Teddy	8.99%	4.07%	1.32	0.012	3.53
Cones	4.66%	2.47%	1.08	0.008	4.01

TABLE 6. The comparison of the accuracy of depth maps estimated using the proposed method and other methods tested in Middlebury Stereo Evaluation Version 3 [52].

Depth estimation method	Percentage of bad pixels for Teddy		Average error for Teddy
	$E_T = 2.0$	$E_T = 4.0$	
DISCO [74]	6.55%	2.73%	0.84
iResNet [75]	12.0%	5.48%	1.22
Fen-D2DRR_ROB [76]	8.76%	5.35%	1.30
Proposed method	8.99%	4.07%	1.32

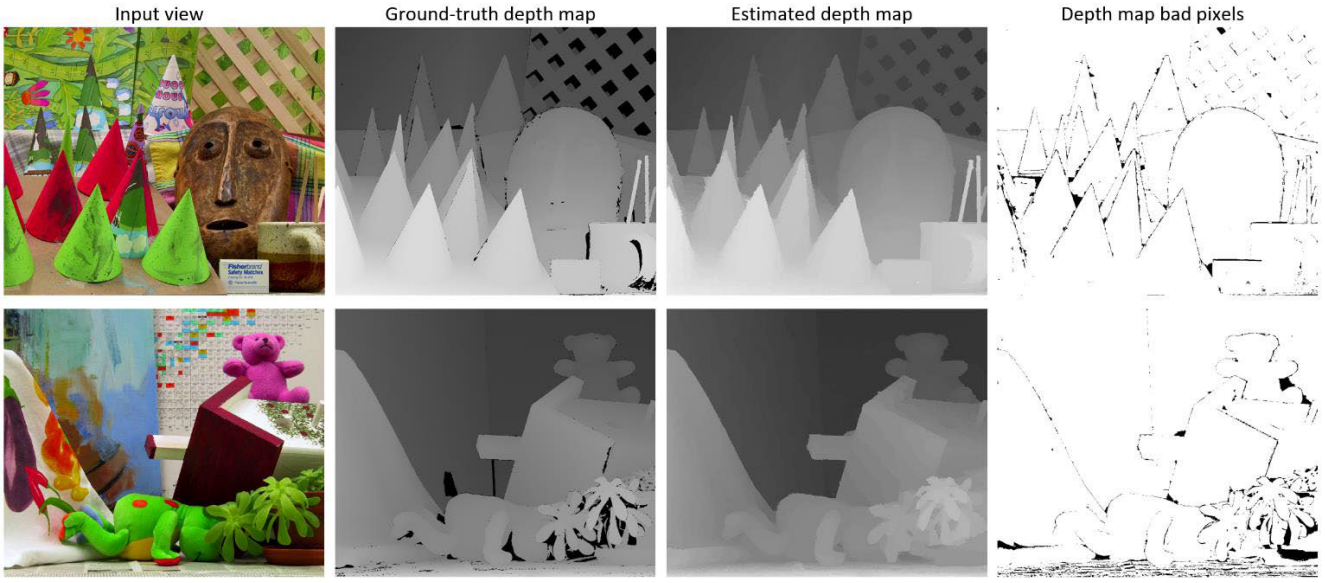

FIGURE 13. Input views of Cones and Teddy sequences with ground-truth depth maps, depth maps estimated with the proposed method and images of bad pixels ($E_T = 4.0$, white color indicates correctly estimated depth).

TABLE 7. The quality of virtual views synthesized using depth maps estimated for different numbers of segments.

Test sequence	Number of segments						
	1,000	5,000	10,000	25,000	50,000	100,000	150,000
	Mean PSNR of the virtual views [dB]						
Ballet	26.55	28.02	28.30	28.53	28.64	28.83	28.85
Breakdancers	29.12	30.85	31.40	31.67	32.00	32.14	32.15
BBB Butterfly	30.26	32.07	32.36	32.79	33.08	33.23	33.25
BBB Rabbit	24.84	26.08	26.64	26.91	27.14	27.43	27.73
Poznań Blocks	22.43	24.55	25.27	25.74	26.60	27.14	27.26
Poznań Blocks2	25.94	26.93	27.16	27.54	27.87	28.10	28.19
Poznań Fencing2	26.17	27.29	27.60	27.92	28.19	28.43	28.61
Poznań Service2	24.35	24.66	24.80	24.93	25.17	25.38	25.51
Average:	26.21	27.55	27.94	28.25	28.59	28.84	28.94

TABLE 8. The quality of virtual views synthesized using depth maps estimated for different numbers of views used in estimation.

Test sequence	Number of views					
	3	4	5	6	7	8
	Mean PSNR of the virtual views [dB]					
Ballet	28.68	28.78	28.63	28.93	28.98	29.01
Breakdancers	31.98	32.05	31.99	32.24	32.14	32.28
BBB Butterfly	32.32	33.13	33.08	33.63	33.62	33.55
BBB Rabbit	26.82	27.25	27.13	27.60	27.27	27.44
Poznań Blocks	25.95	26.37	27.13	26.32	26.66	26.66
Poznań Blocks2	28.14	28.27	28.09	27.73	27.94	27.89
Poznań Fencing2	28.12	28.15	28.42	28.64	28.68	28.72
Poznań Service2	25.14	25.27	25.37	25.06	25.03	25.02
Average:	28.39	28.66	28.73	28.77	28.79	28.82

TABLE 9. The quality of virtual views synthesized using depth maps estimated for different parallelization types.

Test sequence	Parallelization type						
	None	Interleaved levels of depth			Blocks of depth levels		
	Number of threads used in depth estimation						
	1	2	4	6	2	4	6
	Mean PSNR of the virtual views [dB]						
Ballet	28.64	28.64	28.72	28.71	28.34	28.30	28.18
Breakdancers	32.00	32.05	32.06	31.98	31.93	31.87	31.87
BBB Butterfly	33.08	33.09	32.95	32.85	33.20	33.13	33.19
BBB Rabbit	27.14	27.11	27.10	27.07	27.04	26.97	27.02
Poznań Blocks	27.14	26.67	26.33	26.47	27.12	27.08	27.04
Poznań Blocks2	28.10	28.07	28.01	27.97	28.10	28.09	28.10
Poznań Fencing2	28.43	28.38	28.22	28.22	28.41	28.36	28.36
Poznań Service2	25.38	25.39	25.40	25.36	25.27	25.27	25.22
Average:	28.74	28.68	28.60	28.58	28.68	28.63	28.62

The results are summarized in Table 5. We present the percentage of bad pixels of the estimated depth maps summarized over for all available pixels of ground-truth depth maps for the error threshold of $E_T = 2.0$ and $E_T = 4.0$ (i.e., if the absolute error of estimated depth value for a pixel is larger than E_T then this pixel is considered as a bad pixel), an average error, an average relative error and RMSE. Fig. 13 shows the estimated depth maps together with corresponding input view, ground-truth depth maps and visualizations of bad pixels for $E_T = 4.0$.

The proposed method achieves a very low average error of estimated depth maps (on average slightly larger than 1 for 256 depth map levels), which indicates a very high accuracy of estimated depth maps. Current (as of December 2019) top 10 depth estimation methods tested in Middlebury Stereo Evaluation Version 3 [51] for the Teddy sequence achieve the average error smaller than 1.36 and the percentage of bad pixels smaller than 5.57% ($E_T = 4.0$) (e.g. methods described in [72], [73] and [74], see Table 6). Therefore, the proposed method shows state-of-the-art results in terms of the depth maps accuracy.

Nevertheless, what should be stressed again, such evaluation does not measure the inter-view and temporal consistencies of depth maps, crucial for the virtual view synthesis performed in FTV and VN systems. These important features of the proposed method are tested in experiments presented in the previous subsections.

TABLE 10. The quality of virtual views synthesized using depth maps estimated for different numbers of P-Type depth frames between I-type depth frames.

Test sequence	Number of P-type depth frames between I-type depth frames				
	0	4	9	24	49
	Mean PSNR of the virtual views [dB]				
Ballet	28.69	28.68	28.63	28.74	28.75
Breakdancers	32.19	32.13	31.99	31.95	31.75
BBB Butterfly	33.20	33.14	33.08	32.97	32.93
BBB Rabbit	27.21	27.16	27.13	27.12	27.08
Poznań Blocks	27.20	27.19	27.13	26.98	26.79
Poznań Blocks2	28.12	28.11	28.09	28.06	28.03
Poznań Fencing2	28.60	28.50	28.43	28.36	28.35
Poznań Service2	25.51	25.45	25.37	25.19	25.04
Average:	28.84	28.80	28.73	28.67	28.59

TABLE 11. The bitrate and quality of encoded virtual views. Virtual views were synthesized using depth maps estimated for different numbers of P-type depth frames between I-type depth frames.

Test sequence	QP	Number of P-type depth frames between I-type depth frames									
		0		4		9		24		49	
		Bitrate [Mbps]	PSNR [dB]	Bitrate [Mbps]	PSNR [dB]	Bitrate [Mbps]	PSNR [dB]	Bitrate [Mbps]	PSNR [dB]	Bitrate [Mbps]	PSNR [dB]
Ballet	22	4.6	41.8	3.9	41.8	3.7	41.8	3.6	41.8	3.6	41.8
	27	1.7	39.8	1.5	39.9	1.4	39.9	1.4	40.0	1.4	40.0
	32	0.7	37.8	0.6	37.9	0.6	37.9	0.6	37.9	0.6	38.0
	37	0.3	35.9	0.3	35.9	0.3	35.9	0.3	35.9	0.3	35.9
Break-dancers	22	8.7	40.3	8.4	40.4	8.4	40.4	8.1	40.5	8.0	40.5
	27	2.8	38.2	2.7	38.3	2.7	38.3	2.7	38.3	2.6	38.5
	32	1.1	36.4	1.1	36.5	1.1	36.5	1.1	36.5	1.1	36.7
	37	0.5	34.7	0.5	34.8	0.5	34.8	0.5	34.9	0.5	35.1
BBB Butterfly	22	5.5	45.4	5.2	45.4	5.1	45.4	5.1	45.4	5.1	45.4
	27	2.5	42.0	2.4	42.0	2.4	42.1	2.3	42.1	2.3	42.1
	32	1.2	39.1	1.1	39.1	1.1	39.1	1.1	39.1	1.1	39.1
	37	0.5	36.5	0.5	36.5	0.5	36.5	0.5	36.6	0.5	36.5
BBB Rabbit	22	14.8	40.4	10.2	40.8	7.6	41.2	5.5	41.5	4.7	42.0
	27	6.0	36.0	5.0	36.6	3.8	37.1	2.7	37.4	2.3	37.8
	32	2.9	32.7	2.1	33.2	1.6	33.6	1.1	33.9	1.0	34.2
	37	0.9	30.2	0.7	30.5	0.6	30.8	0.4	31.0	0.4	31.2
Poznań Blocks	22	25	41.5	19.2	41.6	18.2	41.7	17.8	41.7	18.2	41.8
	27	9.5	38.1	7.5	38.2	7.0	38.4	6.8	38.3	7.0	38.4
	32	3.5	35.5	2.8	35.6	2.6	35.7	2.5	35.7	2.5	35.7
	37	1.3	33.4	1.0	33.5	0.9	33.6	0.9	33.6	0.9	33.6
Poznań Blocks2	22	26.5	40.1	22.2	40.1	20.8	40.2	19.9	40.2	19.6	40.3
	27	7.7	37.6	6.5	37.7	6.0	37.8	5.7	37.9	5.6	37.9
	32	2.4	35.7	2.1	35.8	1.9	35.9	1.8	35.9	1.8	36.0
	37	0.8	34.1	0.7	34.1	0.6	34.2	0.6	34.2	0.6	34.3
Poznań Fencing2	22	32.3	40.5	30.9	40.8	30.1	40.8	29.8	40.8	29.5	40.8
	27	14.5	36.7	13.4	37.6	12.9	37.7	12.6	37.7	12.4	37.7
	32	5.9	33.4	5.3	34.7	5.1	34.8	4.9	34.8	4.7	34.9
	37	2.0	32.0	1.8	32.5	1.7	32.5	1.7	32.5	1.6	32.7
Poznań Service2	22	46.2	39.7	37.0	39.7	34.7	39.8	35.6	39.8	36.1	39.8
	27	18.1	36.3	14.8	36.5	13.8	36.7	14.4	36.6	14.5	36.6
	32	6.7	33.6	5.7	33.8	5.4	33.9	5.7	33.8	5.7	33.7
	37	2.3	31.6	2.0	31.7	1.8	31.7	2.0	31.5	2.1	31.4

VII. CONCLUSION

The goal of the work is to provide an efficient depth estimation method for applications in FTV and VN. As discussed above, these applications pose specific requirements in addition to the usual expectation of high fidelity and accuracy of depth as well as the pursuit for limited processing time. The FTV/VN applications also require the estimation of several depth maps at a time, temporal and inter-view consistency, and versatility related to arbitrary locations of source cameras.

The approach considered, i.e., segment-based depth estimation has proved to be able to fulfill the abovementioned requirements as demonstrated by the experimental results reported in the paper. The novelty of the approach consists in:

- an original segment-based technique,
- new techniques for temporal and inter-view consistency of depth maps,
- a novel parallelization method.

In the paper, the results are provided that demonstrate the advantages of the proposed method over the Depth Estimation Reference Software (DERS) developed by MPEG. The quality of a depth map is measured by the quality of the synthesized views, and it is higher on average by 2.6 dB. This significant quality improvement with respect to the state-of-the-art DERS is obtained despite the significant reduction of the estimation time by about 4.5 times. The application of the proposed temporal consistency enhancement method increases this reduction to 29 times on average. Moreover, the proposed parallelization results in the reduction of the estimation time up to 130 times with respect to DERS using 6 threads. As there is no commonly accepted measure of the consistency of depth maps, the application of compression efficiency of depth is proposed as a measure of depth consistency. The experimental results are provided in order to demonstrate the quality of depth as functions of segment size, the number of input views used and the number of P-type depth frames.

Although the paper is focused on video, results for still images are also provided that demonstrate that the accuracy of the described method is among the state-of-the-art methods in the Middlebury Stereo Evaluation for multiview static pictures.

A unique feature of the work is related to the disclosure of the source code of the implementation that can be used by other researchers as a new reference for their future works.

The particular usefulness of the presented depth estimation method was already confirmed by its implementation in an operational FTV system developed by the authors from the Chair of Multimedia Telecommunications and Microelectronics of the Poznań University of Technology [56].

APPENDIX

See Tables 7–11.

REFERENCES

[1] G. Lafruit, M. Domański, K. Wegner, T. Grajek, T. Senoh, J. Jung, P. Kovács, P. Goorts, L. Jorissen, A. Munteanu, B. Ceulemans, P. Carballeira, S. García, and M. Tanimoto, "New visual coding exploration in MPEG: Super-multiview and free navigation in free viewpoint TV," in *Proc. Electron. Imag. Conf., Stereoscopic Displays Appl.*, San Francisco, CA, USA, 2016, pp. 1–9.

[2] F. Zilly, C. Riechert, M. Müller, P. Eisert, T. Sikora, and P. Kauff, "Real-time generation of multi-view video plus depth content using mixed narrow and wide baseline," *J. Vis. Commun. Image Represent.*, vol. 25, no. 4, pp. 632–648, May 2014.

[3] L. Fang, Y. Xiang, N.-M. Cheung, and F. Wu, "Estimation of virtual view synthesis distortion toward virtual view position," *IEEE Trans. Image Process.*, vol. 25, no. 5, pp. 1961–1976, May 2016.

[4] Y.-S. Kang and Y.-S. Ho, "High-quality multi-view depth generation using multiple color and depth cameras," in *Proc. IEEE Int. Conf. Multimedia Expo*, Singapore, Jul. 2010, pp. 1405–1410.

[5] S. Xiang, L. Yu, Q. Liu, and Z. Xiong, "A gradient-based approach for interference cancelation in systems with multiple Kinect cameras," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Beijing, China, May 2013, pp. 13–16.

[6] G. Zhang, J. Jia, T.-T. Wong, and H. Bao, "Consistent depth maps recovery from a video sequence," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 6, pp. 974–988, Jun. 2009.

[7] E. Juarez, *Manual of Depth Estimation Reference Software*, Standard ISO/IEC JTC1/SC29/WG11 MPEG W18450, Geneva, Switzerland, 2019.

[8] L. Jorissen, P. Goorts, G. Lafruit, and P. Bekaert, "Multi-view wide baseline depth estimation robust to sparse input sampling," in *Proc. 3DTV-Conf., True Vis.-Capture, Transmiss. Display 3D Video (DTV-CON)*, Hamburg, Germany, Jul. 2016, pp. 1–4.

[9] V. Kolmogorov and R. Zabih, "What energy functions can be minimized via graph cuts?" *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 2, pp. 147–159, Feb. 2004.

[10] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 11, pp. 1222–1239, Nov. 2001.

[11] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High-quality video view interpolation using a layered representation," *TOGACM Trans. Graph.*, vol. 23, no. 3, pp. 600–608, Aug. 2004.

[12] P. Kovacs, *[FTV AHG] Big Buck Bunny Light-Field Test Sequences*, Standardization document ISO/IEC JTC1/SC29/WG11, MPEG M35721, Geneva, Switzerland, 2015.

[13] M. Domański, A. Dziembowski, M. Kurc, A. Łuczak, D. Mieloch, J. Siast, O. Stankiewicz, and K. Wegner, *Poznan University of Technology Test Multiview Video Sequences Acquired with Circular Camera Arrangement-'Poznan Team' and 'Poznan Blocks' Sequences*, Standardization document ISO/IEC JTC1/SC29/WG11, MPEG M35846, Geneva, Switzerland, 2015.

[14] M. Domański, A. Dziembowski, A. Grzelka, D. Mieloch, O. Stankiewicz, and K. Wegner, *Multiview Test Video Sequences for Free Navigation Exploration Obtained Using Pairs of Cameras*, Standardization document ISO/IEC JTC1/SC29/WG11, MPEG M38247, Geneva, Switzerland, 2016.

[15] O. Stankiewicz, K. Wegner, M. Tanimoto, and M. Domański, *Enhanced View Synthesis Reference Software (VSRS) for Free-Viewpoint Television*, Standardization document ISO/IEC group (JTC 1/SC 29/WG 11), MPEG M31520, Geneva, Switzerland, 2013.

[16] P. Yadati and A. M. Namboodiri, "Multiscale two-view stereo using convolutional neural networks for unrectified images," in *Proc. 15th IAPR Int. Conf. Mach. Vis. Appl. (MVA)*, Nagoya, Japan, 2017, pp. 346–349.

[17] J. M. Facil, A. Concha, L. Montesano, and J. Civera, "Single–view and multi–view depth fusion," *IEEE Robot. Autom. Lett.*, vol. 2, no. 4, pp. 1994–2001, Oct. 2017.

[18] B. Tippetts, D. J. Lee, K. Lillywhite, and J. Archibald, "Review of stereo vision algorithms and their suitability for resource-limited systems," *J. Real-Time Image Process.*, vol. 11, no. 1, pp. 5–25, Jan. 2016.

[19] L. Fang, N.-M. Cheung, D. Tian, A. Vetro, H. Sun, and O. C. Au, "An analytical model for synthesis distortion estimation in 3D video," *IEEE Trans. Image Process.*, vol. 23, no. 1, pp. 185–199, Jan. 2014.

[20] R. Achanta and S. Susstrunk, "Superpixels and polygons using simple non-iterative clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 4895–4904.

[21] H. Zhu, Q. Wang, and J. Yu, "Occlusion–model guided antiocclusion depth estimation in light field," *IEEE J. Sel. Top. Signal Process.*, vol. 11, no. 7, pp. 965–978, Oct. 2017.

[22] K. Calagari, M. Elgharib, P. Didyk, A. Kaspar, W. Matusik, and M. Hefeeda, "Data driven 2-d-to-3-d video conversion for soccer," *IEEE Trans. Multimedia*, vol. 20, no. 3, pp. 605–619, Mar. 2018.

[23] L. Li, S. Zhang, X. Yu, and L. Zhang, "PMSC: Patchmatch–based superpixel cut for accurate stereo matching," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 3, pp. 679–692, Mar. 2018.

[24] N. Vretos and P. Daras, "Temporal and color consistent disparity estimation in stereo videos," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Paris, France, Oct. 2014, pp. 3798–3802.

[25] M. F. Tappen and W. T. Freeman, "Comparison of graph cuts with belief propagation for stereo, using identical MRF parameters," in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, Nice, France, Oct. 2003, pp. 900–906.

[26] H. Shi, H. Zhu, J. Wang, S.-Y. Yu, and Z.-F. Fu, "Segment-based adaptive window and multi-feature fusion for stereo matching," *J. Algorithms Comput. Technol.*, vol. 10, no. 1, pp. 3–11, Mar. 2016.

[27] M. Sizintsev and R. P. Wildes, "Spatiotemporal stereo and scene flow via stequel matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 6, pp. 1206–1219, Jun. 2012.

[28] W. Chen, M.-J. Zhang, and Z.-H. Xiong, "Fast semi-global stereo matching via extracting disparity candidates from region boundaries," *IET Comput. Vis.*, vol. 5, no. 2, pp. 143–150, Mar. 2011.

[29] O. Stankiewicz, M. Domanski, A. Dziembowski, A. Grzelka, D. Mieloch, and J. Samelak, "A free–viewpoint television system for horizontal virtual navigation," *IEEE Trans. Multimedia*, vol. 20, no. 8, pp. 2182–2195, Aug. 2018.

[30] M. Domański, O. Stankiewicz, K. Wegner, and T. Grajek, "Immersive visual media—MPEG-I: 360 video, virtual navigation and beyond," in *Proc. IEEE Int. Conf. Syst., Signals Image Process. IWSSIP*, Poznań, Poland, May 2017, pp. 1–9.

[31] Y. Zhang, H. Lv, Y. Liu, H. Wang, X. Wang, Q. Huang, X. Xiang, and Q. Dai, "Light–field depth estimation via epipolar plane image analysis and locally linear embedding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 4, pp. 739–747, Apr. 2017.

[32] D. Mieloch, A. Dziembowski, A. Grzelka, O. Stankiewicz, and M. Domanski, "Graph-based multiview depth estimation using segmentation," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Hong Kong, Jul. 2017, pp. 217–222.

[33] G. Nur, S. Dogan, H. K. Arachchi, and A. M. Kondoz, "Impact of depth map spatial resolution on 3D video quality and depth perception," in *Proc. 3DTV-Conf., True Vis.-Capture, Transmiss. Display 3D Video*, Tampere, Finland, 2017, pp. 1–4.

[34] W. Liu, X. Chen, J. Yang, and Q. Wu, "Robust color guided depth map restoration," *IEEE Trans. Image Process.*, vol. 26, no. 1, pp. 315–327, Jan. 2017.

[35] T. Emori, M. P. Tehrani, K. Takahashi, and T. Fujii, "Free-viewpoint video synthesis from mixed resolution multi-view images and low resolution depth maps," *Proc. SPIE, Stereoscopic Displays Appl. XXVI*, vol. 9391, Mar. 2015, Art. no. 93911C.

[36] M. Camplani, T. Mantecon, and L. Salgado, "Depth–color fusion strategy for 3-D scene modeling with kinect," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 1560–1571, Dec. 2013.

[37] J. Hernandez-Aceituno, R. Arnay, J. Toledo, and L. Acosta, "Using kinect on an autonomous vehicle for outdoors obstacle detection," *IEEE Sensors J.*, vol. 16, no. 10, pp. 3603–3610, May 2016.

[38] A. Dziembowski, A. Grzelka, D. Mieloch, O. Stankiewicz, K. Wegner, and M. Domański, "Multiview synthesis—Improved view synthesis for virtual navigation," in *Proc. Picture Coding Symp. (PCS)*, Nuremberg, Germany, 2016, pp. 1–5.

[39] K. Müller, P. Merkle, and T. Wiegand, "3-D video representation using depth maps," *Proc. IEEE*, vol. 99, no. 4, pp. 643–656, Apr. 2011.

[40] M. Tanimoto, "FTV standardization in MPEG," in *Proc. 3DTV-Conf., True Vis.-Capture, Transmiss. Display 3D Video (DTV-CON)*, Budapest, Hungary, 2014, pp. 1–4.

[41] M. Tanimoto, M. P. Tehrani, T. Fujii, and T. Yendo, "FTV for 3-D spatial communication," *Proc. IEEE*, vol. 100, no. 4, pp. 905–917, Apr. 2012.

[42] C. Lee, A. Tabatabai, and K. Tashiro, "Free viewpoint video (FVV) survey and future research direction," *APSIPA Trans. Signal Inf. Process.*, vol. 4, p. e15, Oct. 2015.

[43] M. Domański, A. Dziembowski, D. Mieloch, A. Łuczak, O. Stankiewicz, and K. Wegner, "A practical approach to acquisition and processing of free viewpoint video," in *Proc. Picture Coding Symp. (PCS)*, Cairns, QLD, Australia, 2015, pp. 10–14.

[44] V. Kolmogorov and R. Zabih, "Multi-camera scene reconstruction via graph cuts," in *Proc. 7th Eur. Conf. Comput. Vis. III (ECCV)*, London, U.K., 2002, pp. 82–96.

[45] O. Stankiewicz, M. Domański, and K. Wegner, "Estimation of temporally-consistent depth maps from video with reduced noise," in *Proc. 3DTV-Conf., True Vis.-Capture, Transmiss. Display 3D Video (DTV-CON)*, Lisbon, Portugal, 2015, pp. 1–4.

[46] *HEVC Reference Codec*. Accessed: Mar. 18, 2019. [Online]. Available: https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/

[47] G. Bjøntegaard, *Calculation of Average PSNR Differences Between RD986 Curves*, Standardization document ISO/IEC group (JTC 1/SC 29/WG 11), MPEG M15378, Austin, TX, USA, 2001.

[48] S. Li, C. Zhu, and M.-T. Sun, "Hole filling with multiple reference views in DIBR view synthesis," *IEEE Trans. Multimedia*, vol. 20, no. 8, pp. 1948–1959, Aug. 2018.

[49] Z. Lee and T. Q. Nguyen, "Multi–array camera disparity enhancement," *IEEE Trans. Multimedia*, vol. 16, no. 8, pp. 2168–2177, Dec. 2014.

[50] X. Suau, J. Ruiz-Hidalgo, and J. R. Casas, "Real–time head and hand tracking based on 2.5D data," *IEEE Trans. Multimedia*, vol. 14, no. 3, pp. 575–585, Jun. 2012.

[51] D. Scharstein, H. Hirschmüller, Y. Kitajima, G. Krathwohl, N. Nesic, X. Wang, and P. Westling, "High-resolution stereo datasets with subpixel-accurate ground truth," in *Proc. German Conf. Pattern Recognit. (GCPR)*, Münster, Germany, 2014, pp. 31–42.
[52] T. Schöps, J. L. Schönberger, S. Galliani, T. Sattler, K. Schindler, M. Pollefeys, and A. Geiger, "A multi-view stereo benchmark with high-resolution images and multi-camera videos," in *Proc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, 2017, pp. 2538–2547.
[53] T. Senoh, N. Tetsutani, and H. Yasuda, "Depth estimation and view synthesis for immersive media," in *Proc. Int. Conf. 3D Immersion (ICD)*, Brussels, Belgium, 2018, pp. 1–8.
[54] Y. Chang, S. Kim, and Y. Ho, "Depth upsampling methods for high resolution depth map," in *Proc. Int. Conf. Electron., Inf., Commun. (ICEIC)*, Honolulu, HI, USA, 2018, pp. 1–4.
[55] X. Jiang, M. L. Pendu, and C. Guillemot, "Depth estimation with occlusion handling from a sparse set of light field views," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Athens, Greece, Oct. 2018, pp. 634–638.
[56] M. Domański, A. Dziembowski, T. Grajek, A. Grzelka, K. Klimaszewski, D. Mieloch, R. Ratajczak, O. Stankiewicz, J. Siast, J. Stankowski, and K. Wegner, "Demonstration of a simple free viewpoint television system," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Beijing, China, Sep. 2017, pp. 4589–4591.
[57] O. Stankiewicz, G. Lafruit, and M. Domański, "Multiview video: Acquisition, processing, compression and virtual view rendering," in *Image and Video Processing and Analysis and Computer Vision*, vol. 6, R. Chellappa and S. Theodoridis, Eds. New York, NY, USA: Academic, 2018, pp. 3–74.
[58] X. Huang, J. Zhang, Q. Wu, L. Fan, and C. Yuan, "A coarse-to-fine algorithm for matching and registration in 3D cross–source point clouds," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 10, pp. 2965–2977, Oct. 2018.
[59] Y. Pan, R. Liu, B. Guan, Q. Du, and Z. Xiong, "Accurate depth extraction method for multiple light-coding-based depth cameras," *IEEE Trans. Multimedia*, vol. 19, no. 4, pp. 685–701, Apr. 2017.
[60] *Overview of 3D Video Coding*, Standardization document ISO/IEC JTC1/SC29/WG11, MPEG N9784, Archamps, France, May 2008.
[61] J. Lei, L. Li, H. Yue, F. Wu, N. Ling, and C. Hou, "Depth map super–resolution considering view synthesis quality," *IEEE Trans. Image Process.*, vol. 26, no. 4, pp. 1732–1745, Apr. 2017.
[62] G. Lee, B. Li, and C. Chen, "Content-adaptive depth map enhancement based on motion distribution," in *Proc. IEEE Vis. Commun. Image Process. Conf.*, Valletta, Malta, Dec. 2014, pp. 482–485.
[63] L. Hong and G. Chen, "Segment-based stereo matching using graph cuts," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun./Jul. 2004.
[64] O. Stankiewicz, M. Domański, and K. Wegner, "Estimation of temporally-consistent depth maps from video with reduced noise," in *Proc. 3DTV-Conf., True Vis.-Capture, Transmiss. Display 3D Video (DTV-CON)*, Lisbon, Portugal, 2015, pp. 1–4.
[65] T. Xue, A. Owens, D. Scharstein, M. Goesele, and R. Szeliski, "Multi-frame stereo matching with edges, planes, and superpixels," *Image Vis. Comput.*, vol. 91, Nov. 2019, Art. no. 103771.
[66] C. Yao, T. Tillo, Y. Zhao, J. Xiao, H. Bai, and C. Lin, "Depth map driven hole filling algorithm exploiting temporal correlation information," *IEEE Trans. Broadcast.*, vol. 60, no. 2, pp. 394–404, Jun. 2014.
[67] O. Stankiewicz, "Noise in multiview videos," in *Noise Reduction: Methods, Applications and Technology*, M. Farrow, Ed. Hauppauge, NY, USA: Nova Science Publishers, 2018, pp. 1–33.
[68] O. Stankiewicz, M. Domański, and K. Wegner, "Analysis of noise in multi-camera systems," in *Proc. DTV Conf.*, Budapest, Hungary, vol. 2014.
[69] S. Xu, F. Zhang, X. He, X. Shen, and X. Zhang, "PM–PM: Patchmatch with potts model for object segmentation and stereo matching," *IEEE Trans. Image Process.*, vol. 24, no. 7, pp. 2182–2196, Jul. 2015.
[70] G. Wu, Y. Li, Y. Huang, and Y. Liu, "Joint view synthesis and disparity refinement for stereo matching," *Frontiers Comput. Sci.*, vol. 13, no. 6, pp. 1337–1352, Dec. 2019.
[71] D. Scharstein and R. Szeliski, "High-accuracy stereo depth maps using structured light," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Madison, WI, USA, vol. 1, Jun. 2003, pp. 195–202.
[72] K. Swami, K. Raghavan, N. Pelluri, R. Sarkar, and P. Bajpai, "DISCO: Depth inference from stereo using context," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Shanghai, China, Jul. 2019, pp. 502–507.
[73] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "FlowNet 2.0: Evolution of optical flow estimation with deep networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 1647–1655.
[74] X. Ye, J. Li, H. Wang, H. Huang, and X. Zhang, "Efficient stereo matching leveraging deep local and context information," *IEEE Access*, vol. 5, pp. 18745–18755, 2017.

DAWID MIELOCH received the M.Sc. and Ph.D. degrees from the Poznań University of Technology, in 2014 and 2018, respectively. He is currently an Assistant Professor with the Chair of Multimedia Telecommunications and Microelectronics. He is actively involved in ISO/IEC MPEG activities, where he contributes to the development of the immersive media technologies. He has been involved in several projects focused on multiview and 3-D video processing. His professional interests include free viewpoint television, depth estimation, and camera calibration.

OLGIERD STANKIEWICZ received the M.Sc. and Ph.D. degrees from the Faculty of Electronics and Telecommunications, Poznań University of Technology, in 2014. He is currently an Assistant Professor with the Chair of Multimedia Telecommunications and Microelectronics. He is actively involved in ISO standardization activities, where he contributes to the development of the 3-D video coding standards. From 2011 to 2014, he was a Coordinator of the development of MPEG reference software for 3-D-video coding standards based on AVC. Now, he contributes to MPEG free viewpoint TV and JPEG-PLENO standardization activities. He has published over ninety MPEG/JPEG standardization documents and about thirty articles on free view television, depth estimation, view synthesis, and hardware implementation in FPGA. His professional interests include signal processing, video compression algorithms, computer graphics, and hardware solutions. In 2005, he received the Second Place in the IEEE Computer Society International Design Competition (CSIDC), held in Washington, DC, USA.

MAREK DOMAŃSKI received the M.Sc., Ph.D., and Habilitation degrees from the Poznań University of Technology, Poland, in 1978, 1983, and 1990, respectively. Since 1993, he has been a Professor with the Poznań University of Technology, where he leads the Chair of the Multimedia Telecommunications and Microelectronics. He has coauthored one of the very first AVC decoders for TV set-top boxes, in 2004, and highly ranked technology proposals to MPEG for scalable video compression, in 2004, and 3-D video coding, in 2011. He has authored three books and over 300 articles in journals and conference proceedings. The contributions were mostly on image, video and audio compression, virtual navigation, free-viewpoint television, image processing, multimedia systems, 3-D video and color image technology, digital filters and multidimensional signal processing. He has served as a member of various steering, program, and editorial committees of international journals and international conferences. He was the General Chairman/Co-Chairman and host of several international conferences: Picture Coding Symposium, PCS 2012; IEEE International Conference Advanced and Signal Based Surveillance, AVSS 2013; European Signal Processing Conference, EUSIPCO 2007; 73rd and 112nd Meetings of MPEG; International Workshop on Signals, Systems and Image Processing, IWSSIP 1997 and 2004; International Conference Signals and Electronic Systems, ICSES 2004; and others.